CLAIM REVISIONS

1. (previously presented) A method of determining the flow of a data object in a software architecture using queues to organize the transfer of data from one processing object to another, comprising :

storing a queue indicator in a path object corresponding to a respective data object;

receiving and processing the data object in a first of said processing objects;

identifying a queue corresponding to a second of said processing objects the identifying depending on the indicator in the path object corresponding to said data object;

placing said data object in the queue identified in said step of identifying.

2. (previously presented) A method as in claim 1, wherein said step of identifying includes determining a result of said step of processing.

3. (previously presented) A method as in claim 1, wherein: said step of identifying includes determining a result of said step of processing; and said queue corresponding to said result.

4-5. (cancelled or withdrawn)

6. (previously presented) A pipeline software architecture in which data objects are transferred from a first processing object to a selected one of second and third processing objects by queuing the data objects in a queue of said selected one, comprising:

a path object corresponding to each of said data objects;

at least one of said path objects containing an indicator of at least one of said second and

CLAIM REVISIONS

third processing object;

said first processing object defining a process a result of which is to insure that a first data object processed by said first processing object is placed in a queue of said at least one of said second and third processing objects responsively to one of said path objects corresponding to said first data object.

7. (previously presented) An architecture as in claim 6, wherein said process includes the generation of an indication of a result of processing of said first processing object and said first data object processed by said first processing object is placed in said queue of said at least one of said second and third processing objects responsively to the processing object indicator in the at least one of said path objects corresponding to said first data object and responsively to said result indication.

8-9. (canceled)

10. (previously presented) The method of claim 1, wherein the path object includes a table of queue indicators.

11. (currently amended) A method of determining the flow of a data object in a software architecture using queues to organize the transfer of data from one processing object to another, comprising :

storing a queue indicator in a path object corresponding to a respective data object;

CLAIM REVISIONS

receiving and processing the data object in a first of said processing objects;

identifying a queue corresponding to a second of said processing objects the identifying depending on the indicator in the path object corresponding to said data object;

placing said data object in the queue identified in said step of identifying~~The method of claim 1,~~

wherein

the processing comprises determining a normal or faulty outcome state of the data object; and

the identifying is dependent on said normal or faulty outcome state.

12. (previously presented) A method comprising:

defining objects, each comprising both data and functions that access the data, the objects including: data objects, and path objects and processing objects;

first queuing a data object in a queue of a first processing object in response to a [1]indication of the first processing object in a path object associated with the data object;

responsive to the first queuing, processing the data object with the first processing object;

second queuing the data object in a queue of a second processing object in response to both: results of the processing; and an indication of the second processing object in the path object associated with the data object;

responsive to the second queuing, processing the data object with a second processing object.

[1] sic, should be "an."

CLAIM REVISIONS

13. (previously presented) Apparatus comprising:

objects, each object comprising both data and functions that access the data, the objects including: data objects and path objects and processing objects, each path object mutually corresponding to a respective data object;

a respective processing queue for each processing object, the processing objects each process each data object previously queued in the respective queue, the processing of the data object including using the functions of the data object to access the data of the data object, the path objects each comprising indicators of next processing objects for subsequent processing of the corresponding data object after the processing of the data object by the current processing object, the current processing object communicating with the path object to determine the next processing objects for subsequent processing[2] the data object, after the processing of the data object by the current processing unit is complete the processing object queues the data object in a queue of one of the next processing objects depending on a result of the processing of the data object by the current processing object.

[2] Sic, should be followed by "of"